# ICARO

Nuestro compromiso es con el ***futuro***.

# Back End

# Clase 4

# ¿Qué veremos hoy?

Hoy vamos a continuar con el módulo de **Back-End**, donde veremos una introducción al patrón de diseño MVC.

Continuaremos mejorando nuestras rutas e iremos introduciendo nuevos conceptos como el manejo de parámetros. Finalmente veremos la estructura final de nuestro proyecto a fin de ir poniendo en práctica estos conceptos.

ICARO

# Patron MVC. Significado

¿Qué es el patrón MVC?

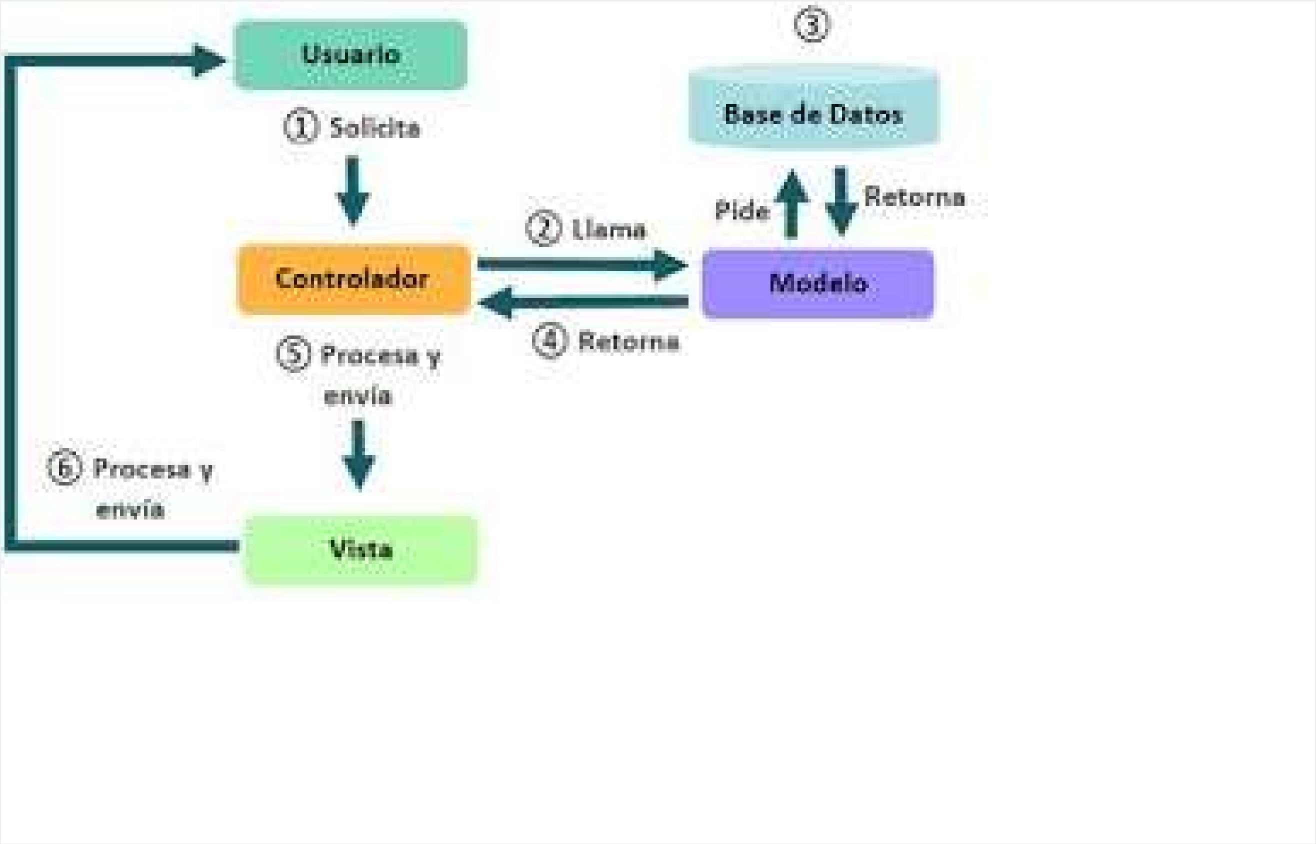
**Modelo**: Representa la lógica de negocios. Es el encargado de acceder de forma directa a los datos actuando como “**intermediario**” con la fuente de datos.

**Vista**: Es la encargada de mostrar la información al usuario de forma gráfica y “humanamente legible”.

**Controlador**: Es el intermediario entre la vista y el modelo. Es quien controla las interacciones del usuario solicitando los datos al modelo y entregándolos a la vista para que ésta, lo presente al usuario.

ICARO

# Patron MVC. Diagrama



# Rutas y el objeto ‘Request’

Anteriormente vimos que en nuestras definiciones de rutas hacemos uso de dos objetos propios de EXPRESS un objeto para el manejo de peticiones o REQUEST y otro para el manejo de las respuestas o RESPONSE como usualmente se lo podrá encontrar en documentaciones.

```
app.get('/', (req, res) => {

    res.render('home')

})
```

# Rutas y el objeto 'Request'

En EXPRESS podemos utilizar varios elementos pertenecientes al objeto request para la manipulación de la data, veamos los más útiles y su explicación correspondiente:

**req.params**

Este objeto nos permite obtener las rutas, nombres y sus parámetros, se utiliza para obtener más que todos los datos que envía una sección de nuestra aplicación.

Bien, vimos una introducción a el patrón MVC y cómo mejorar nuestras rutas; ahora aplicaremos todo !.

ICARO

# Rutas. Especificando parametros

Vamos a suponer que ingresamos en nuestro navegador http://localhost:3000/usuarios/1 , como resultado esperamos obtener una vista que por ahora solamente nos muestre la informacion de ese usuario en particular. La forma de lograr ese resultado es mediante el uso del objeto **req.params.**

Luego de la palabra usuarios, nuestra ruta, como esta planteada, espera un valor, ese valor puede ser cualquier número el cual iremos obteniendo de forma dinámica dentro de nuestra aplicación.

# ¡Vamos al código!

# Estructura del proyecto. MVC. Rutas

Como habiamos mencionado, queremos que en base a nuestra ruta se muestre una vista diferente, mas precisamente, queremos que al especificar un ‘id’ de usuario distinto la vista cambie. Para esto necesitamos que exista una fuente de datos a la que ir a consultar si existe ese usuario y así poder obtener sus datos. De momento esto estará presente en nuestra aplicación sin necesidad de consultar ninguna fuente externa (base de datos por ejemplo).

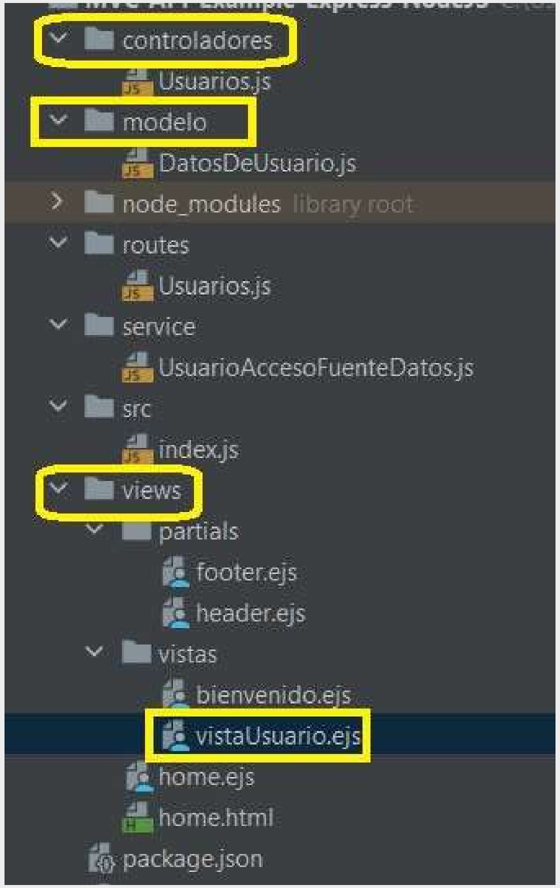
ICARO

# Estructura del proyecto.

Vemos planteada la estructura, acorde al patron MVC.
Respecto al modelo como comentamos, en este caso, vemos
Un archivo de nombre DatosDeUsuario.js, solo lo tomemos como
una fuente temporal para acceder a dos usuarios distintos.

Vemos a nuestro controlador, nuestras ya conocidas vistas y para
este caso usaremos nuestra vista para mostrar datos de usuario;
De momento seguiremos mostrando nombre de usuario.

Nuestras vistas, seguirán usando EJS como ya hemos visto. Tanto
Express como EJS estan especificados como dependencias en
El package.json



# Resultado esperado de nuestro proyecto

Ingresamos http://locahost:3000/usuarios/1 y vemos

Si cambiamos el id e ingresamos http://localhost:3000/usuarios/2 obtendriamos:

# Tips. Ejecución del proyecto

Recordar siempre hacer uso de nuestro gestor de dependencias npm a la hora de iniciar nuestro proyecto. Nuestro package.json siempre deberá estar actualizado y por ende tendremos que ejecutar el correspondiente npm install a fin de que nuestro proyecto funcione.

**No olvidemos concurrir a clases de consulta!!!!**

# Muchas gracias!

ICARO Asociación Civil
CUIT 30716564815
info@icaro.org.ar
www.icaro.org.ar